A COMMON FRAMEWORK FOR ANALYSIS AND GENERATION

Allan Ramsay
Department of Computer Science,
University College Dublin,
Belfield, DUBLIN 4, Ireland

ABSTRACT

It seems highly desirable to use a single representation of linguistic knowledge for both analysis and generation. We argue that the only part of the average NL system's knowledge that we can have any faith in is its vocabulary and, to a lesser extent, its syntactic rules, and we investigate the consequences of this for generation.

1 ANALYSIS

Consider a typical NLU system. You give it a piece of text, say:

(1) *The house I live in is damp.*

It grinds away, trying out syntactic rules until it has an analysis of the structure of the text. The syntactic rules incorporate a semantic element, which automatically builds up a representation of the meaning of the text in some appropriate formal language — something like the following: $\mathit{presupp}(\bullet[\exists!\ B\ (\mathit{house}(B)\ \&\ \mathit{presupp}(\bullet[\exists\ !\ C\ (\mathit{speaker}(C))]),\ \exists\ D\ (\mathit{state}(D,\ \mathit{live})\ \&\ \mathit{agent}(D,\ C)\ \&\ \exists\ E : \{\mathit{interval}(E)\}\ (\mathit{contains}(E,\ \mathit{now})\ \&\ \mathit{during}(E,\ D))\ \&\ \mathit{in}(D,\ B)))]),\ \exists\ F\ (\mathit{condition}(F,\ \mathit{damp})\ \&\ \mathit{object}(F,\ B)\ \&\ \exists\ G : \{\mathit{interval}(G)\}\ (\mathit{contains}(G,\ \mathit{now})\ \&\ \mathit{during}(G,\ F)))$

Exactly what formal language you choose for the representation of meaning will depend on a number of things, notably on the intended application (if any) of the system, on the availability of automatic inference systems for the language in question, and on the perceived need for expressive power. For the system that lies behind the discussion in this paper we chose a version of Turner's [1987] property theory. The details of property theory do not really matter very much here. What matters is that any attempt to give a complete formal paraphrase of (1) must include at least as much information as we have given above. In particular, the logical structure of our paraphrases contains essential information (about, for instance, the differences between objects which are introduced in the utterance and ones whose existence is presupposed), even if there is still considerable debate about the best way of representing this information.

2 GENERATION

Suppose we have the formula given above as a formal paraphrase of (1), and we want to generate an English sentence which corresponds to it. We might hope to use our syntactic/semantic rules "backwards", looking for something which would generate a sentence and whose semantic component could be made to match the given sequence. The final rule we actually used in our analysis of (1) is an elaboration of the standard S → NP VP rule which contains a description of how the meanings of the NP and the VP should be combined to obtain the meaning of the NP. Space does not permit inclusion of this rule. The important point for our present purposes is that the representation of the meaning of the S is built up from the discourse representations of the subject and the predicate. The subject and predicate each provide some background constraints, and then their meanings get combined (along with a complex abstraction to the effect that there is some object E which satisfies two properties P0 and P1) to produce a further constraint. The question we want to investigate here is: can we use rules of this kind to generate (1) from the above semantic representation?

The problem is that rules of this kind explain how to combine the meanings of constituents *once you have identified them.* Given an expression of property theory like the one above, it is very difficult to see how to decompose into parts corresponding to an NP and a VP. So difficult, in fact, that without a great deal of extra guidance it must be regarded as impossible.

The final semantic representation reflects our beliefs about the best formal paraphrase of the English text, whereas the semantic representations of the components reflect the way we think that this paraphrase might be obtained. Somebody else might decide that they liked our final analysis, but that they preferred some other way of deriving it. In view of the number of different ways of obtaining a given expression E as the result of simplifying some complex expression ($t \in \bullet[x, P]$), it is simply unreasonable to hope to find the right decomposition of a given semantic representation unless you already know a great deal about the way the linguistic theory being used builds up its representations. Indeed, unless you already have this knowledge it is unlikely that you will even be able to tell whether some semantic representation has a realisation as a natural language text at all.

If we look again at the knowledge available to our "average NL system", we see that it will include a vocabulary of lexical items, a set of syntactic rules, and a set of semantic interpretations of those rules. It is worth reflecting briefly on the evidence that lies behind particular choices of lexical entry, grammatical rule and semantics interpretation.

The evidence that leads to a particular choice of words to go in the vocabulary is fairly concrete. We can, for instance, take a corpus of written English and collect all the contiguous sequences of letters separated by spaces. We can be fairly confident that nearly every such sequence is a word, and that those things that are not words will be fairly easily detected. We would in fact probably want to do a bit better than simply collecting all such letter sequences, since we would want to recognise the connection between *eat* and *eaten*, and between *die* and *dying*, but at least the objects that we are interested in are available for inspection.

The evidence that leads to a particular choice of syntactic theory is less directly available. Once we have a vocabulary derived from some corpus, we can start to build up word classes on the basis of looking for words that can be exchanged without turning a meaningful sentence into a meaningless one — to spot that almost any meaningful sentence containing the word *walk* could be turned into a meaningful sentence containing the word *run*, for instance. We can then start looking for phrase types and for relations between phrase types. We can perhaps be reasonably confident about our basic classification into word classes, though we may find some surprises, but the evidence for specific phrase types is often in the eye of the beholder, and the evidence for subtler relationships can be remarkably intangible. Nonetheless, there is some concrete evidence, and it has led to some degree of consensus about the basic elements of syntactic theory. You will, for instance, find very few NL systems that do not utilise the notion of an NP, or that do not recognise the phenomena of agreement and unbounded dependency.

The evidence for specific semantic theories, by contrast, is almost entirely circumstantial. We can usually tell whether two sentences mean the same thing; we can usually tell whether a sentence is ambiguous; and we can sometimes tell whether one sentence entails another, or whether one contradicts another. To get from here to a decision that one representation scheme is more appropriate than another, and to a particular translation of some piece of NL into the chosen scheme, requires quite a bit of faith. In order to build a system for translating NL input into some computer-amenable representation we have no choice but to make that act of faith. We have to choose a representation scheme, and we have to decide how to translate specific fragments of NL into it and how to combine such translated fragments to build translations of larger fragments. Examples abound. The system that constructed the translation of (1) into the given sequence of propositions in PT is described and defended at length in [Ramsay 1990], and we will not recapitulate it here. We note, however, that the rules we use for translating from English into this representation scheme will not generate arbitrary such sequences. Only sequences which correspond to the output of the rules we are using applied to the translations we have allocated to the lexical items in our vocabulary will be generated. *This is true of all NL systems that translate from a natural language into some formal representation language.* For any such system, only a fraction of the possible sentences of the representation language will correspond to direct translations of NL sentences, and the only way of telling which they are is to look for the corresponding NL sentence.

Suppose we wanted to develop a system which used our linguistic knowledge base to generate texts corresponding to the output of some application system. It would be absurd to expect the application program to generate sentences of our chosen representation language, and to try to work from these via our syntactic/semantic rules to an NL realisation. We have no convincing evidence that our representation language is correct; we have no easy way of specifying which sentences of the representation language correspond via our rules to NL sentences; and even if we did have a sentence in the representation language which corresponded to an NL sentence, we would have a great deal of difficulty in breaking it into appropriate components, particularly if this involved replacing a single formula by the instantiation of some abstraction with an appropriate term.

We suggest instead that the best way to get an NL system to generate text to satisfy the requirements of some application program is for it to offer suggestions about how it is going to build the text, along with explanations of why it is going to build it that way. We therefore supplement our descriptions of linguistic structures with a component describing their functional structure.

For the rule for S, for instance, we add an element describing what the SUBJECT and PRED are for. We could say that the SUBJECT is the theme and the PRED is the rheme, using terms from functional grammar [Halliday 1985] for the purpose. A language generation system using the above rule can now ask the application program whether it is prepared to describe a theme and a rheme. Admittedly this still presumes that the application program knows enough about the linguistic theory to know about themes and rhemes, but at least it does not need to know how they are organised into sentences, how they can be realised, or how their semantic representations are combined to form a sentence in the representation language. Furthermore, if the application progíam is to make full use of the expressive power of NL then it must be able to make sensible choices about such matters, since any hearer will be sensitive to them. If the combination of application program and NL generation system cannot make rational decisions about whether to say, for instance, *John ate it* or *It was eaten by John* then they must expect to be misunderstood by native English speakers who are, albeit unconsciously, aware that these two carry different messages.

Once the application program has agreed to describe a theme and a rheme, the NL system can then elicit these descriptions. Since the rule being used specifies that the theme must be an NP then it can move on to rules and lexical entries that can be used for constructing NPs and start asking questions about these.

3 COMPARISONS

We are concerned here almost entirely with what has come to be known as the "tactical" component of language generation — with how to realise some chosen message as NL text, rather than with how to decide what message we want realised. The two are not entirely separable, but we have little to say about "strategic" tasks such as deciding what properties should be used for characterising an item being referred to by an NP, which we expect the application program to deal with. The responsibility for deciding whether to pronominalise something, for instance, would be handed over to the application program by the NL system bluntly asking whether a description with the property `qualifier:pronoun` was acceptable. We thus completely side-step the issues addressed by systems which plan what to say to produce specific effects in a hearer [Appelt 1985], which work out how organise multi-sentence texts in order to convey complex messages without disorienting the hearer [McKeown 1985], or which invent effective descriptions for use in referring expressions [Dale 1988]. These are all important tasks, but they are not what we are concerned with here.

The most direct comparison is with [Shieber et al. 1990], where an approach to generating text from a given logical form is described. The algorithm described by Shieber and his colleagues takes a realisable λ-calculus expression and uses their syntactic/semantic rules "backwards" to generate appropriate text. Their emphasis is on controlling the way these rules are applied, with rules satisfying certain rather stringent criteria being applied top-down and all other rules being used bottom-up. The algorithm looks effective, so long as (a) it is reasonable to assume that an application program can be relied on to produce realisable expressions in the representation language and (b) there are any rules which satisfy their criteria. We argued at some length above that the first of these conditions is unlikely to hold unless the application program knows a great deal about the syntactic/semantic rules which are going to be used. We also suspect that the way they control the top-down application of rules imposes unacceptable constraints on the way that semantic representations of wholes are composed out of semantic representations of parts. Certainly none of the rules we used in the system described in [Ramsay 1990] satisfy their criteria. We therefore believe that our approach, where the application decides whether the fragments of text proposed by the NL system are acceptable as they are proposed, is more flexible than any approach which depends on getting a realisable expression of the representation language from the application program and systematically translating it into a natural language using syntactic/semantic rules which were primarily designed for translating in the other direction.

REFERENCES

Appelt D. (1985): *Planning English Sentences*: Cambridge University Press, Cambridge.

Dale R. (1988): *Generating Referring Expressions in a Domain of Objects and Processes*, Ph.D. thesis, Centre for Cognitive Science, University of Edinburgh.

Halliday M.A.K. (1985): *An Introduction to Functional Grammar*: Arnold, London.

Kamp H. (1984): A Theory of Truth and Semantic Representation, in *Formal Methods in the Study of Language* (eds. J. Groenendijk, J. Janssen & M. Stokhof): Foris Publications, Dordrecht: 277–322.

McKeown K. (1985): *Generating English Text*: Cambridge University Press, Cambridge.

Ramsay A.M. (1990): *The Logical Structure of English: Computing Semantic Content*: Pitman, London.

Shieber S.M, van Noord G., Pereira F.C.N. & Moore R.C. (1990): Semantic-Head-Driven Generation, *Computational Linguistics* 16(1): 30–42.

Turner R. (1987): A Theory of Properties, *Journal of Symbolic Logic* 52(2): 455–472.